

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

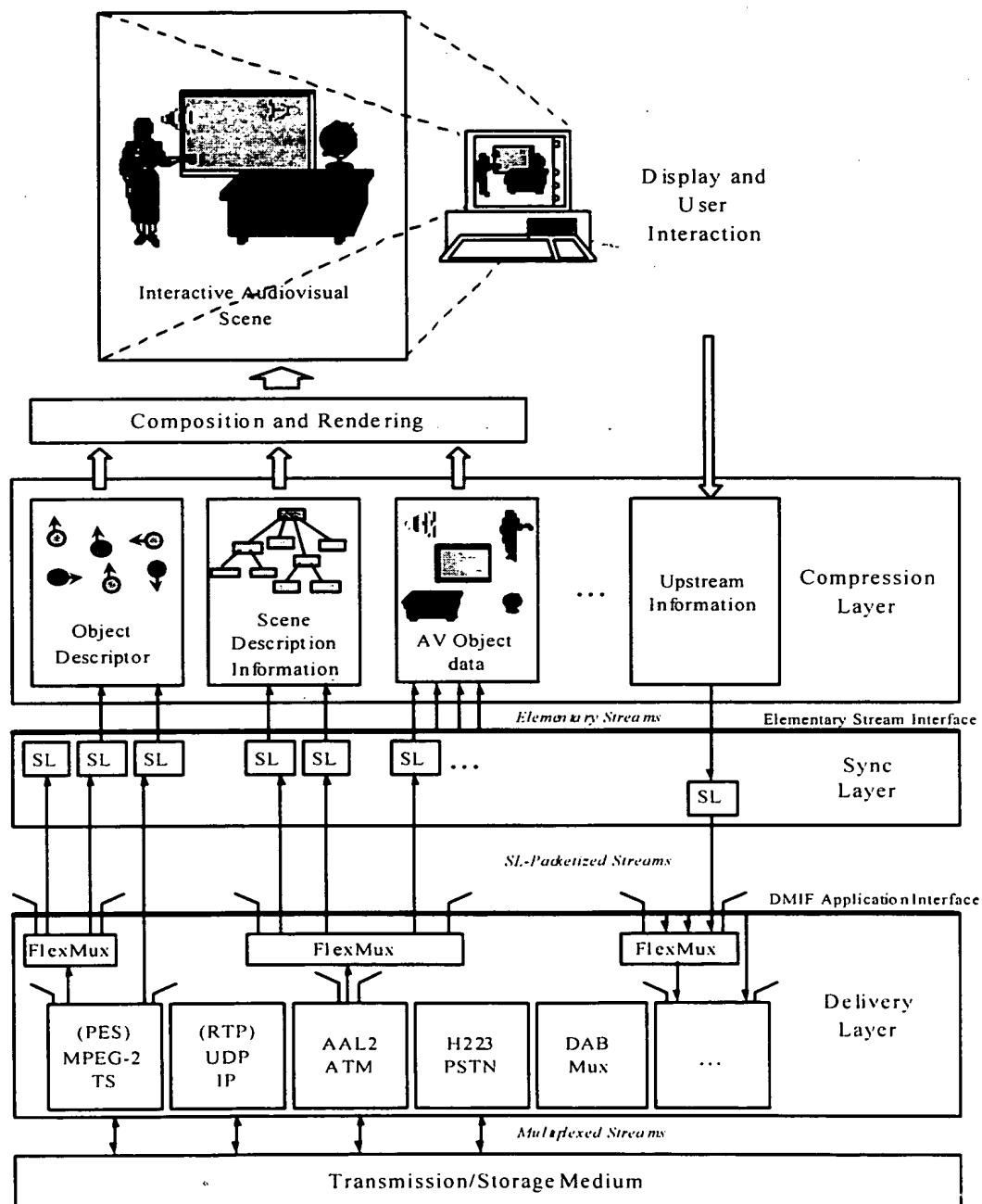


FIG. 1

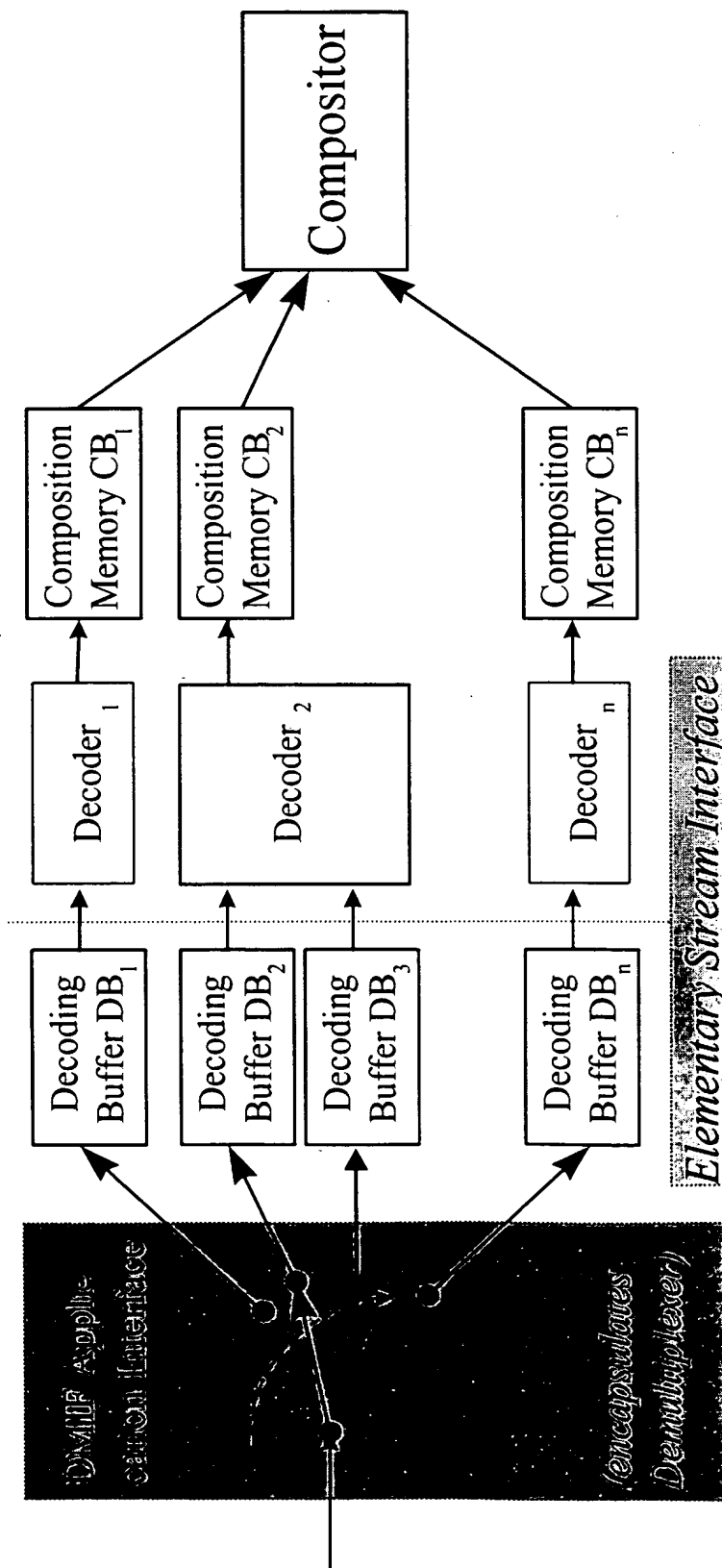


FIG. 2

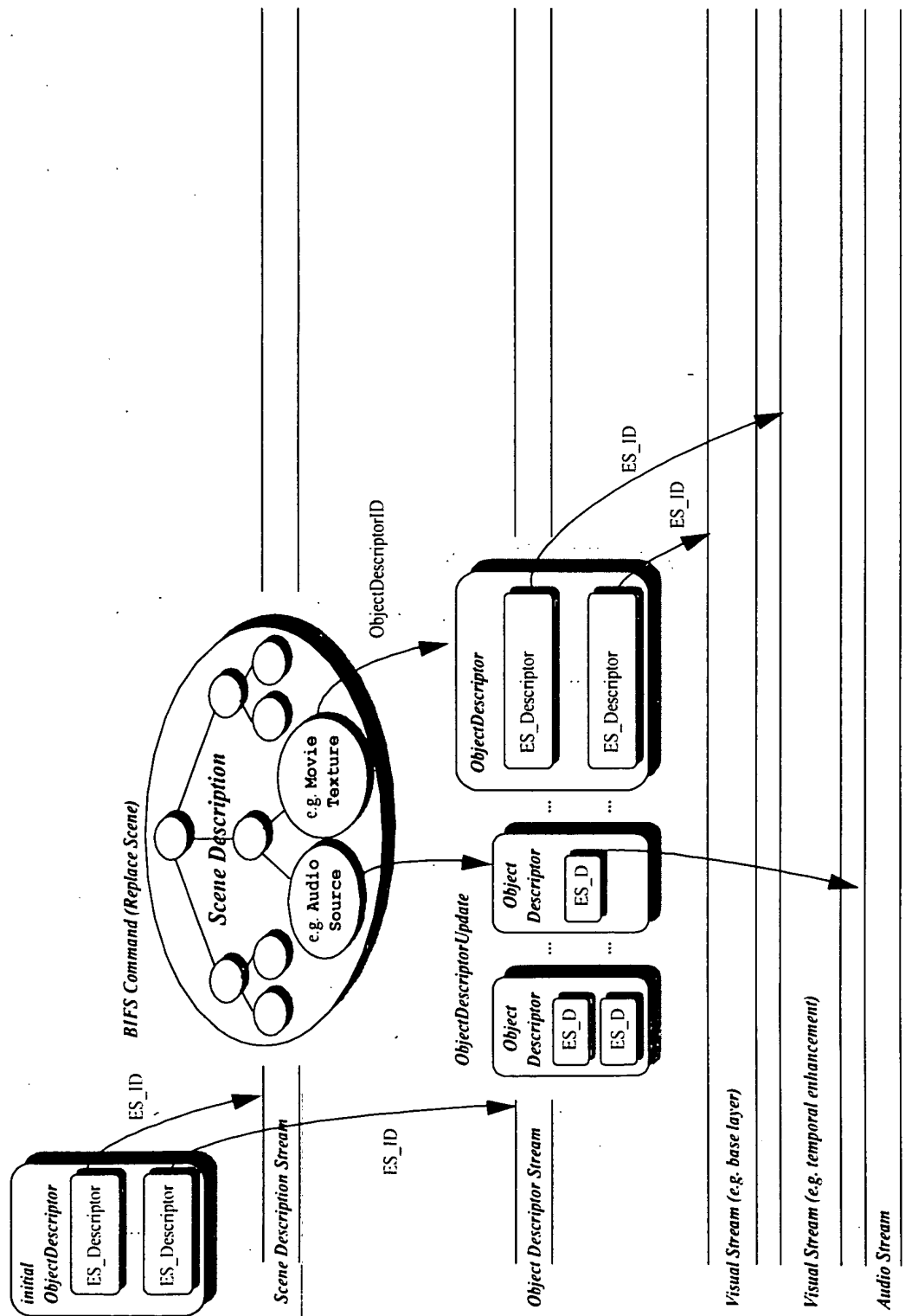


FIG. 3

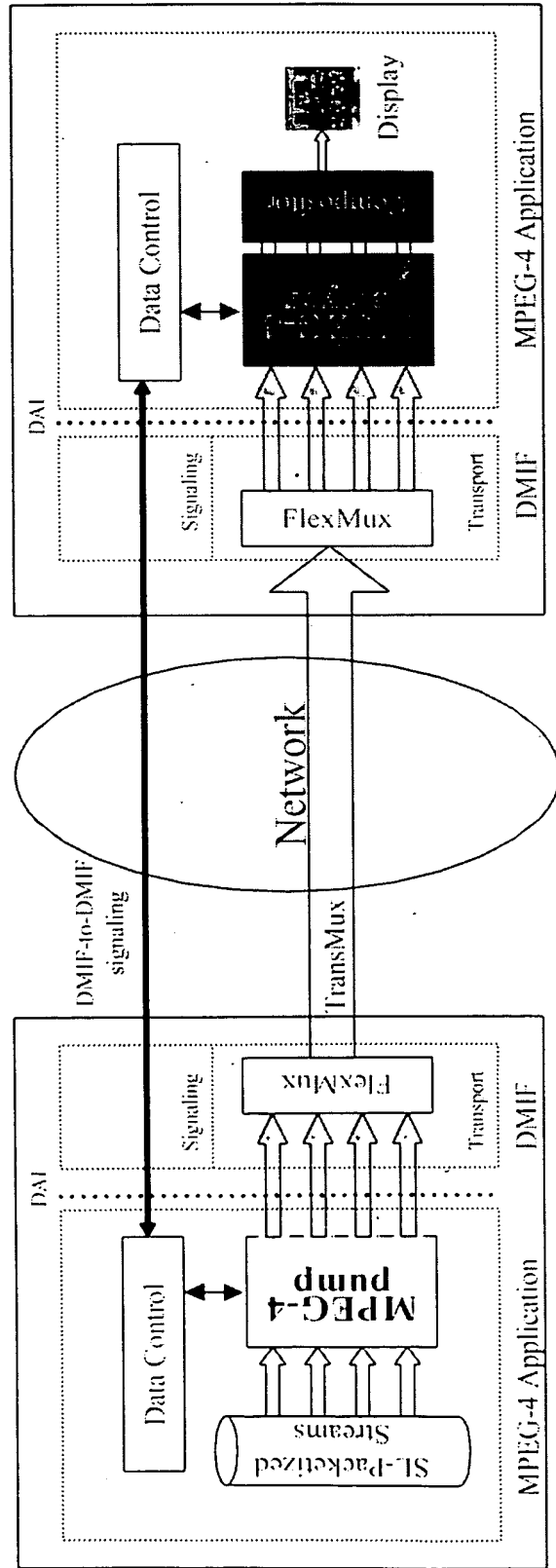
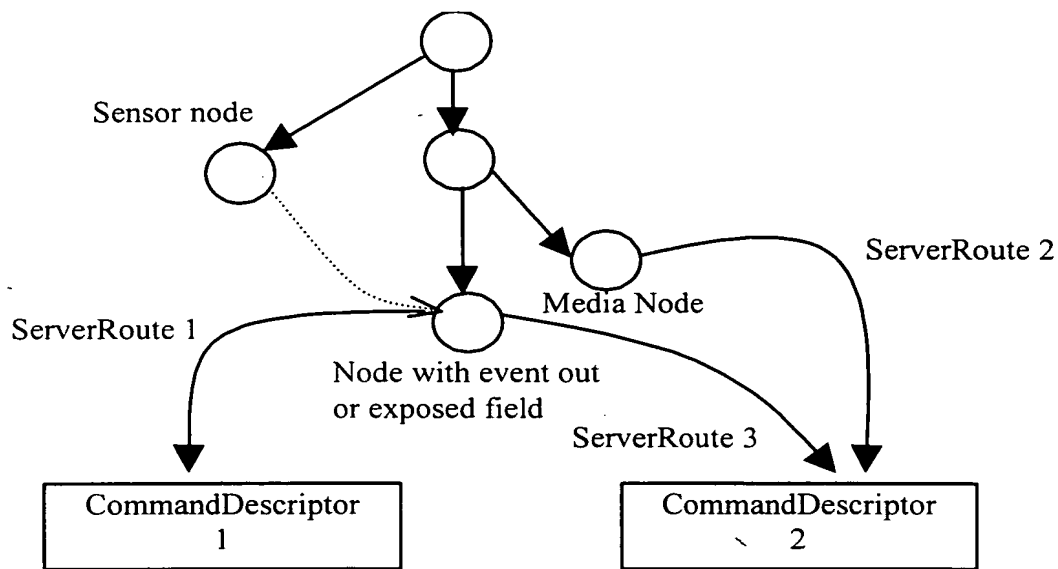
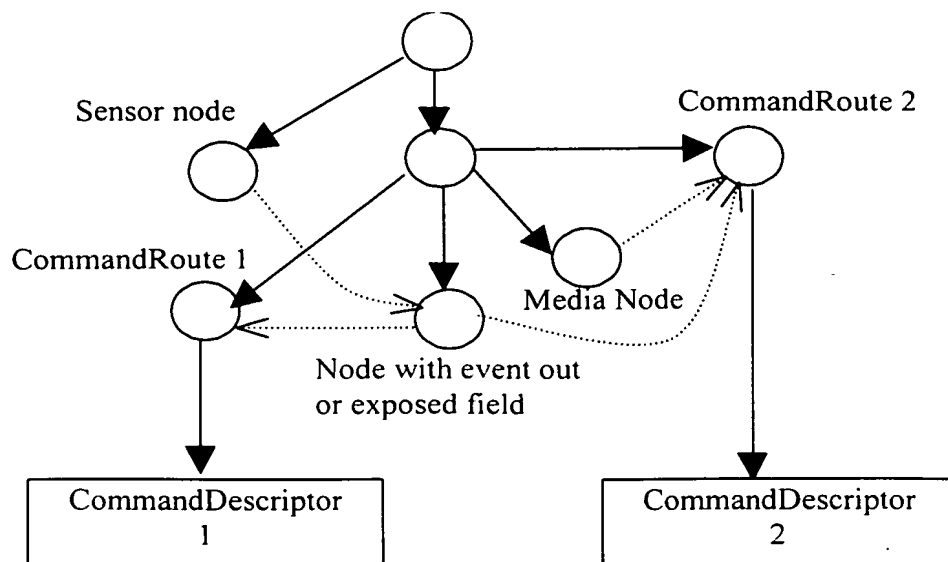


FIG. 4



(a)



(b)

FIG. 5

```

class CommandDescriptor: bit(8) CommandDescriptorTag = 0x05 {
    bit(16) CommandDescriptorID;
    bit(16) CommandID;

    bit(16) length;

    // stream count; number of ES_IDs associated with this message
    unsigned int (8) count;

    // ES_Id (channel numbers) of the streams affected by the command
    unsigned int (16) ES_ID[count];

    // application-defined parameters
    do {
        unsigned int (8) paramLength;
        char (8) commandParam [paramLength];
    }
    while (paramLength!=0);
}

```

FIG. 6

```

class CommandDescriptorRemove: bit(8) CommandDescriptorRemoveTag = 0x06 {
    bit(16) CommandDescriptorID;
}

```

FIG. 7

```

class BIFSScene {
    SFNode nodes(SFTopNode);
    bit(1) hasROUTES;
    if (hasROUTES)
        ROUTEs routes ;
    bit(1) hasServerROUTES;
    // the following is added to the MPEG-4 syntax
    if (hasServerROUTES)
        ServerROUTES sroutes;
}

// modification of MPEG-4 ROUTEs structure to point to command descriptor
class ServerROUTES {

    bit(1) isUpdateable;
    if (isUpdateable)
        bit(10) srouteID;

    bit(10) outNodeID;
    NodeData nodeOUT = GetNodeFromID(outNodeID); // get source node
    int(nodeOUT.nOUTbits) outFieldRef; // event source field index

    bit(10) CD_ID; // event sink - command descriptor ID
}

```

FIG. 8

CommandRoute

Node interface

```
CommandRoute {  
    eventIn      SFBool      Execute      FALSE  
    field        SFUrl       CommandDescriptor  []  
}
```

Functionality and semantics

The **CommandRoute** node is used to support server interaction. A command route is executed when an event is received on the **execute** field, for example, from a touch sensor. The execution of a command route involves communicating the command pointed by the **commandDescriptor** to the server. The **commandDescriptor** field contains either a URL to the command descriptor or the ID of the Command Descriptor to be associated with this **CommandRoute** node. Commands are typically sent to a server using DMIF's DAI_User_Command primitives. The node update mechanism can be used to change the command descriptor ID. This allows supporting different interaction behavior for the same user interaction at different times (before and after node update).

FIG. 9

Command Name	Command ID
Unused	0X0000
Start	0X0001
Stop	0X0002
Pause	0X0003
Forward	0X0004
Reverse	0X0005
ContentSelection	0X0006
MPEG-4 reserved	0X0007 - 0X00FF
User Defined	0X0100 - 0XFFFF

FIG. 10


```

class Command {

    // command ID
    bit(16) CommandID;

    // the following are simply copied from the command descriptor
    // stream count; number of ES_IDs associated to this message
    unsigned int (8) count;

    // ES_Id of the streams
    unsigned int (16) ES_ID[count];

    // application-defined parameters
    do {
        unsigned int (8) paramLength;
        char (8) commandParam [paramLength];
    }
    while (paramLength!=0);
}

```

FIG. 11

2025 RELEASE UNDER E.O. 14176

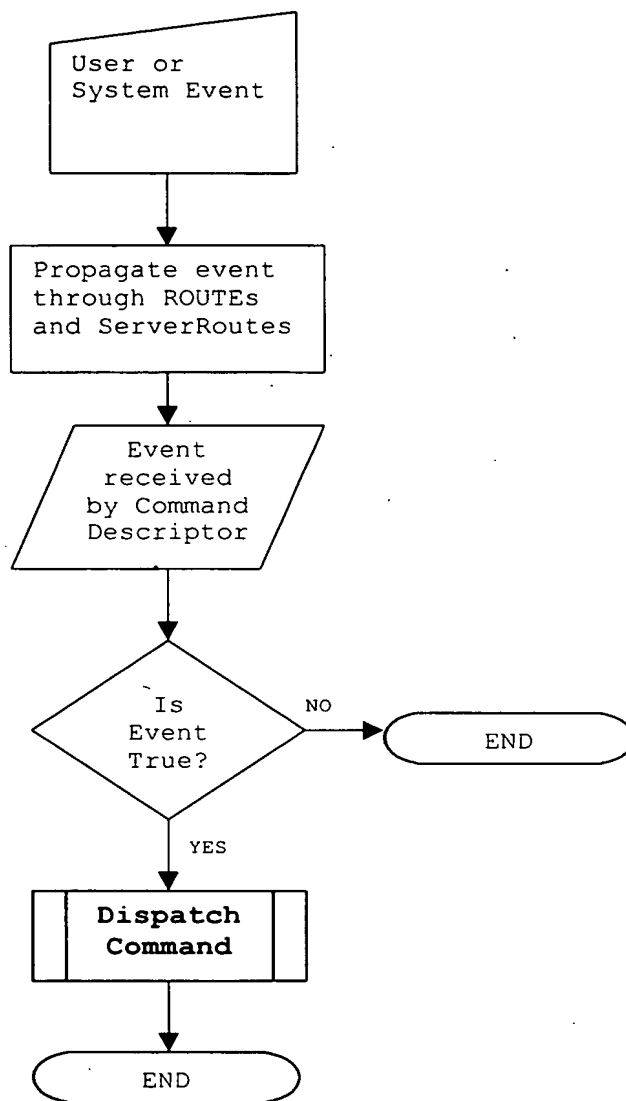


FIG. 12

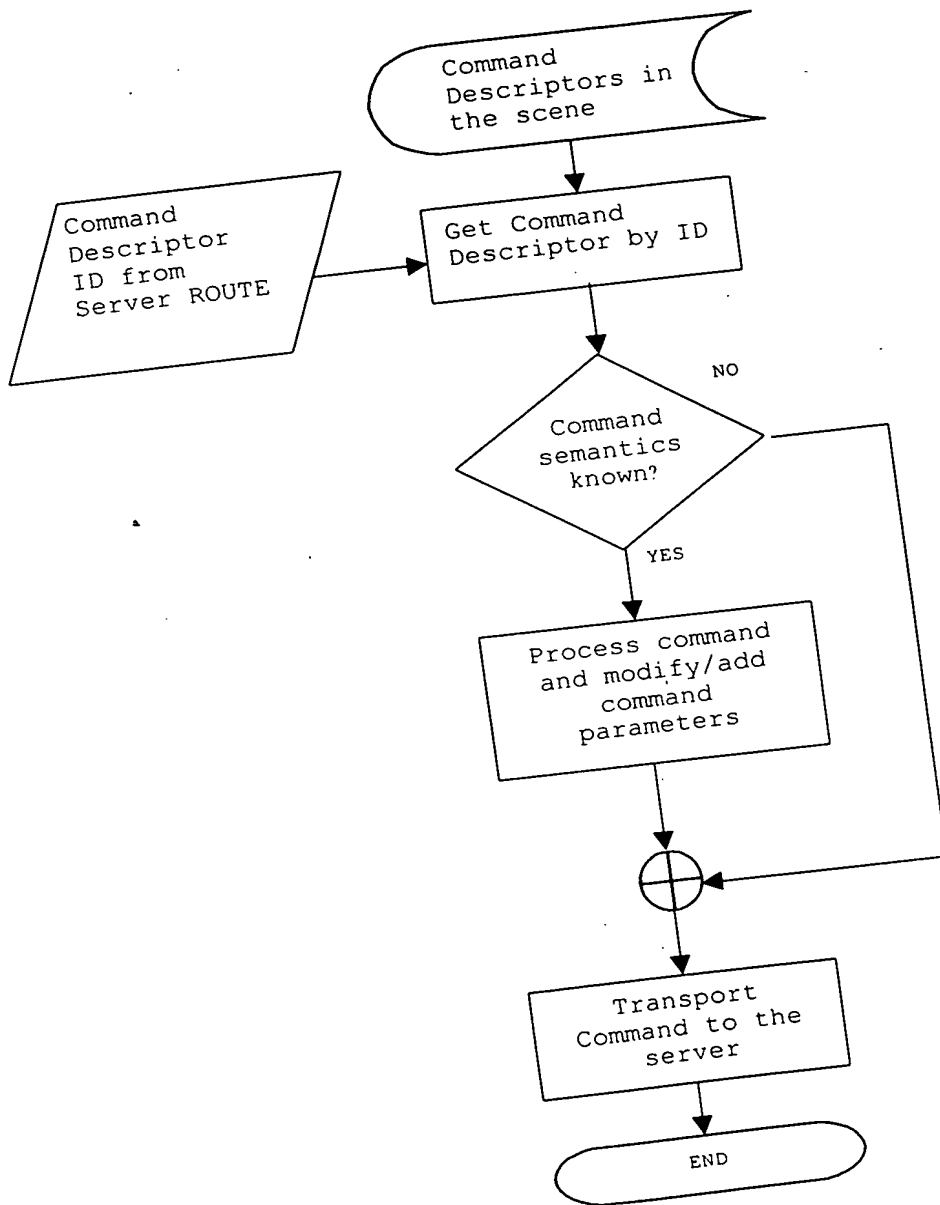


FIG. 13

```

graph TD
    A[User or System Event] --> B[Propagate event through ROUTES]
    B --> C[/Event received by 'execute' field of Command Route node/]
    C --> D{Is Event True?}
    D -- NO --> E([END])
    D -- YES --> F[Dispatch Command]
    F --> G([END])
  
```

FIG. 14

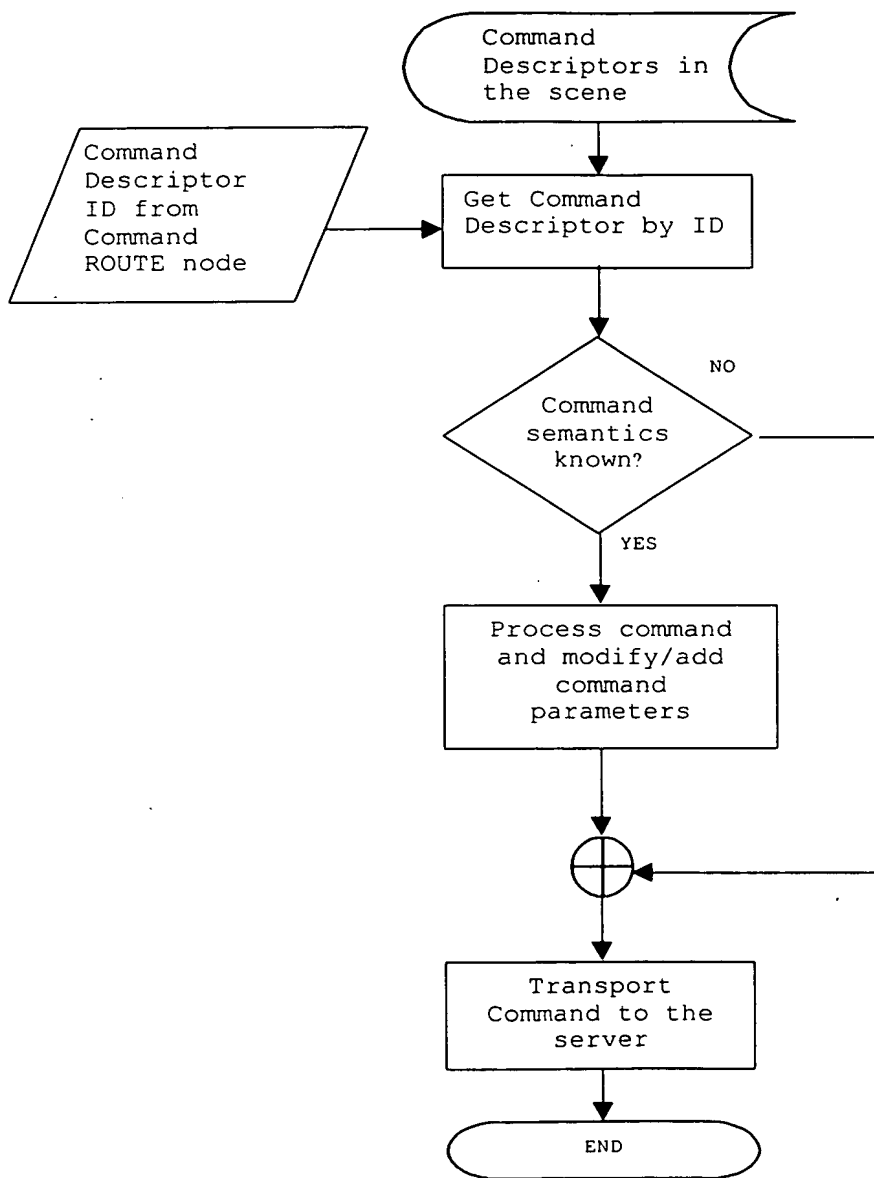


FIG. 15